ON THE DESIGN OF GRADIENT ALGORITHMS EMPLOYING ORTHOGONAL MATRIX CONSTRAINTS

Scott C. Douglas

Department of Electrical Engineering
Southern Methodist University
Dallas, Texas 75275 USA

ABSTRACT

Algorithms for adapting orthogonal matrices in optimization and signal processing typically employ the geometry of either the Grassmann manifold or the Stiefel manifold depending on the chosen cost function. In this paper, we develop gradient adaptive algorithms that use the geometry of both manifolds in their operation. Such algorithms offer a straightforward way to mitigate numerical error accumulation due to discretization of the coefficient updates. Examples drawn from subspace tracking and eigenvector analysis illustrate the usefulness of the design methods.

Index Terms— Grassmann manifold, orthogonality constraints, Stiefel manifold, subspace tracking.

1. INTRODUCTION

Consider the following task: Compute a real-valued $(m \times n)$, $m \leq n$ matrix $\mathbf{W}$ to

$$\text{maximize} \quad \mathcal{J}(\mathbf{W}) \qquad (1)$$

$$\text{such that} \quad \mathbf{W}\mathbf{W}^T = \mathbf{I}, \qquad (2)$$

where $\mathcal{J}(\mathbf{W})$ is a cost function. This general optimization problem is important for numerous problems in numerical linear algebra and signal processing, including subspace tracking and independent component analysis [1]–[9]. The above task induces a useful geometry to the parameter space known as the *Grassmann manifold* for row homogeneous cost functions in which $\mathcal{J}_G(\mathbf{Q}_m\mathbf{W}) = \mathcal{J}_G(\mathbf{W})$ for all $\mathbf{Q}_m$ and the *Stiefel manifold* for row inhomogeneous cost functions in which $\mathcal{J}_S(\mathbf{Q}_m\mathbf{W}) \neq \mathcal{J}_S(\mathbf{W})$ for some $\mathbf{Q}_m$, where $\mathbf{Q}_m$ is an $(m \times m)$ orthonormal and invertible matrix ($\mathbf{Q}_m\mathbf{Q}_m^T = \mathbf{Q}_m^T\mathbf{Q}_m = \mathbf{I}$). By employing Riemannian geometry, adaptive procedures can be developed for solving (1)–(2) that have useful properties, such as computational simplicity or fast convergence; see [1]–[9] for examples.

When designing iterative algorithms to solve (1)–(2) especially for tracking applications, numerical effects can cause a loss of system performance and even divergence of the matrix estimate $\mathbf{W}_k$. This issue has led to the development of *ad hoc* stabilization methods involving additional correction terms within the coefficient updates [3, 8], or projection methods to impose (2) at each iteration [6, 10]–[12]. Geometrical insight into these numerical difficulties has not been given in the scientific literature, nor has any general strategy for stabilizing such algorithms been described for arbitrary tasks.

In this paper, we consider the design of simple gradient adaptive algorithms for solving (1)–(2) for inhomogeneous cost functions that largely avoids numerical difficulties in algorithm implementation. The key idea in this study is to mathematically represent $\mathbf{W}$ as

$$\mathbf{W} = \mathbf{U}\mathbf{V}, \qquad (3)$$

where the $(m \times m)$ matrix $\mathbf{U}$ and $(m \times n)$ matrix $\mathbf{V}$ are adapted on the Stiefel and Grassmann manifolds, respectively. This representation enables one to design simple, efficient numerical stabilization methods for $\mathbf{W}_k$ as well as solve optimization problems that would otherwise be challenging to solve using (1)–(2). Numerical examples drawn from subspace tracking and eigenvector analysis illustrate the usefulness of the methods.

2. MATHEMATICAL PRELIMINARIES

Consider the orthonormal matrix $\mathbf{W}$ at time t. The differential gradient update of $\mathbf{W}$ that attempts to maximize $\mathcal{J}_S(\mathbf{W})$ in the Stiefel manifold is given by

$$\frac{d\mathbf{W}}{dt} = \mathbf{W}\mathbf{W}^T\mathbf{G}_S - \mathbf{W}\mathbf{G}_S^T\mathbf{W}, \qquad (4)$$

where $\mathbf{G}_S = \partial\mathcal{J}_S(\mathbf{W})/\partial\mathbf{W}$ is the Euclidean gradient of the cost function with respect to $\mathbf{W}$ [4]. Eq. (4) maintains

$$\mathbf{W}(t)\mathbf{W}^T(t) = \mathbf{W}(0)\mathbf{W}^T(0) \qquad (5)$$

for all time $t \geq 0$. Therefore, if $\mathbf{W}(0)$ has orthonormal rows, $\mathbf{W}(t)\mathbf{W}^T(t)$ is identity for all time.

The update in (4) can be written as

$$\frac{d\mathbf{W}}{dt} = \mathbf{W}\left[\mathbf{W}^T\mathbf{G}_S - \mathbf{G}_S^T\mathbf{W}\right], \qquad (6)$$

where $\mathbf{W}^T\mathbf{G}_S - \mathbf{G}_S^T\mathbf{W}$ is a skew-symmetric matrix. The multiplicative nature of this update is one of its critical features [9]. Since the skew-symmetric matrix $\mathbf{W}^T\mathbf{G}_S - \mathbf{G}_S^T\mathbf{W}$ multiplies $\mathbf{W}$ on the *right*, this update causes the rows of

$\mathbf{W}$ to rotate with time. As (4) computes geodesics on the Stiefel manifold, no numerical approximation to this computation has been or needs to be assumed at this point.

Consider now the column-wise Stiefel update given by

$$\frac{d\mathbf{W}}{dt} \;=\; \mathbf{G}_S\mathbf{W}^T\mathbf{W} - \mathbf{W}\mathbf{G}_S^T\mathbf{W}. \qquad (7)$$

Comparing (7) with (4), we see that the role of the rows and columns of $\mathbf{W}$ in the constraint space has changed. Eq. (7) maintains the constraint

$$\mathbf{W}^T(t)\mathbf{W}(t) \;=\; \mathbf{W}^T(0)\mathbf{W}(0). \qquad (8)$$

If $\mathbf{W}(0)$ has orthonormal rows, then $\mathbf{W}^T(t)\mathbf{W}(t)$ is a *constant projection matrix* for all $t \geq 0$ for $m < n$. The row span of $\mathbf{W}(t)$ does not change with time using this update.

Consider a homogeneous cost function $\mathcal{J}_G(\mathbf{W})$ to be maximized with respect to $\mathbf{W}$ under the constraint $\mathbf{W}\mathbf{W}^T = \mathbf{I}$. Adaptation of $\mathbf{W}$ in the Grassmann manifold can be performed using

$$\frac{d\mathbf{W}}{dt} \;=\; \mathbf{W}\mathbf{W}^T\mathbf{G}_G - \mathbf{G}_G\mathbf{W}^T\mathbf{W}, \qquad (9)$$

where $\mathbf{G}_G = \partial\mathcal{J}_G(\mathbf{W})/\partial\mathbf{W}$ [4]. This update also maintains $\mathbf{W}\mathbf{W}^T$ as a constant matrix. If $\mathbf{W}(0)\mathbf{W}^T(0) = \mathbf{I}$, then $\mathbf{W}\mathbf{W}^T = \mathbf{I}$ for all time $t \geq 0$, like the algorithm in (4). In such cases, we can write this as the multiplicative update

$$\frac{d\mathbf{W}}{dt} \;=\; \mathbf{W}\left(\mathbf{W}^T\mathbf{G}_G\right)\left[\mathbf{I} - \mathbf{W}^T\mathbf{W}\right], \qquad (10)$$

where $[\mathbf{I} - \mathbf{W}^T\mathbf{W}]$ is a projection matrix.

Note that it makes no sense to consider the analogous algorithm to (7) in the Grassmann manifold. The Grassmann manifold is only appropriate in problems where the orientations of the rows of $\mathbf{W}$ do not matter. A consequence of this fact is that (9) is only useful in situations where $m < n$ such that the linear row span of $\mathbf{W}$ describes an m-dimensional subspace.

3. COMBINED ADAPTATION IN THE GRASSMANN AND STIEFEL MANIFOLDS

The algorithms in (7) and (10) are complementary. Eq. (7) adjusts the orientations of the columns of $\mathbf{W}$ while maintaining $\mathbf{W}^T\mathbf{W}$ as a constant matrix, whereas Eq. (10) adjusts the orientations of the rows of $\mathbf{W}$ while maintaining $\mathbf{W}\mathbf{W}^T$ as a constant matrix. It makes sense to combine these two algorithms. We can derive the combined algorithm by setting $\mathbf{W}(t) = \mathbf{U}(t)\mathbf{V}(t)$, where the $(m \times m)$ matrix $\mathbf{U}(t)$ is adapted using the Stiefel manifold update

$$\frac{d\mathbf{U}}{dt} \;=\; \left[\overline{\mathbf{G}}_S\mathbf{U}^T - \mathbf{U}\overline{\mathbf{G}}_S^T\right]\mathbf{U}, \qquad (11)$$

and the $(m \times n)$ matrix $\mathbf{V}(t)$ is adapted using the Grassmann manifold update

$$\frac{d\mathbf{V}}{dt} \;=\; \mathbf{V}(\mathbf{V}^T\overline{\mathbf{G}}_G)\left[\mathbf{I} - \mathbf{V}^T\mathbf{V}\right]. \qquad (12)$$

In each case, we have computed the gradients

$$\overline{\mathbf{G}}_S \;=\; \frac{\partial\mathcal{J}_S(\mathbf{U}\mathbf{V})}{\partial\mathbf{U}} \;=\; \mathbf{G}_S\mathbf{V}^T \qquad (13)$$

$$\overline{\mathbf{G}}_G \;=\; \frac{\partial\mathcal{J}_G(\mathbf{U}\mathbf{V})}{\partial\mathbf{V}} \;=\; \mathbf{U}^T\mathbf{G}_G. \qquad (14)$$

Thus, we have

$$\frac{d\mathbf{W}}{dt} \;=\; \frac{d\mathbf{U}}{dt}\mathbf{V} + \mathbf{U}\frac{d\mathbf{V}}{dt} \qquad (15)$$

which using (11)–(14) simplifies to

$$\frac{d\mathbf{W}}{dt} = \left[\mathbf{G}_S\mathbf{W}^T - \mathbf{W}\mathbf{G}_S^T\right]\mathbf{W} + \mathbf{W}(\mathbf{W}^T\mathbf{G}_G)[\mathbf{I} - \mathbf{W}^T\mathbf{W}]. \qquad (16)$$

The differential update in (16) has several features:

1. It combines two different differential updates as a linear sum of their respective forms.
2. The first term on the right-hand side of (16) spans the row space of $\mathbf{W}$ at time t.
3. The last term on the right-hand side of (16) moves $\mathbf{W}$ in the null space of $\mathbf{W}$ at time t.
4. The cost functions being optimized for each term do not need to be the same.

This last point is worthy of further justification. To a casual observer, the use of both the Grassmann and Stiefel manifolds may seem redundant. Why bother with adaptation in subspaces when the goal is to identify orthonormal matrices? The answer can be inferred by study of the original Stiefel manifold differential update in (4). Consider the portions of this differential update inside and outside of the current subspace estimate, which we can obtain by multiplying the right-hand side of (4) by the matrix $\left[(\mathbf{W}^T\mathbf{W}) + (\mathbf{I} - \mathbf{W}^T\mathbf{W})\right]$. After simplification, we obtain

$$\frac{d\mathbf{W}}{dt} = \mathbf{G}_S\mathbf{W}^T\mathbf{W} - \mathbf{W}\mathbf{G}_S^T\mathbf{W} + \mathbf{W}\mathbf{W}^T\mathbf{G}_S(\mathbf{I} - \mathbf{W}^T\mathbf{W}). \qquad (17)$$

Comparing (17) and (16), we see that they differ only in the use of different Euclidean gradients in their second terms. In other words, (16) performs adaptation in the Stiefel manifold if the same inhomogeneous cost function is employed both within and outside the current subspace estimate represented by the row span of $\mathbf{W}$.

We can now turn the question around: *Why use the same cost function within and outside the subspace being identified?* Such a restriction is unnecessary given the differential algorithm in (16). We can apply two different costs – corresponding to two different goals – to adjust $\mathbf{W}$ within and outside the subspace represented by the row span of $\mathbf{W}$. The new algorithm preserves the orthonormality of the rows of $\mathbf{W}$ as does the Stiefel manifold update, and it gives an additional design freedom to the user.

4. ALGORITHM DESIGN EXAMPLES

One of the classic tasks in subspace analysis is eigenvector estimation. Given a symmetric positive definite matrix $\mathbf{R}$ of dimension $(n \times n)$ with distinct eigenvalues, find the m eigenvectors corresponding to the m largest or smallest eigenvalues.

One can divide this task into two sub-tasks: (1) identify the m-dimensional subspace spanned by the desired eigenvectors, and (2) find the distinct vectors within this subspace. The update developed in the last section can be used in this regard. Consider the following cost functions:

$$\mathcal{J}_S(\mathbf{W}) = -\frac{1}{2}||\text{tril}[\mathbf{W}\mathbf{R}\mathbf{W}^T]||_F^2 \quad (18)$$

$$\mathcal{J}_G(\mathbf{W}) = \frac{1}{2}\text{tr}[\mathbf{W}\mathbf{R}\mathbf{W}^T], \quad (19)$$

where tril$[\mathbf{M}]$ denotes the strictly lower triangular portion of the square matrix $\mathbf{M}$. The cost $\mathcal{J}_S(\mathbf{W})$ is similar to that used in [9] for eigenvector estimation, and its negative sign leads to minimization of the absolute value of this cost. The cost $\mathcal{J}_G(\mathbf{W})$ is well-known for subspace tracking tasks. It can be shown that $\mathcal{J}_S(\mathbf{W})$ is inhomogeneous and $\mathcal{J}_G(\mathbf{W})$ is homogeneous. The gradients of these cost functions are

$$\mathbf{G}_S = -\text{tril}[\mathbf{W}\mathbf{R}\mathbf{W}^T]\mathbf{W}\mathbf{R} \quad (20)$$

$$\mathbf{G}_G = \mathbf{W}\mathbf{R}. \quad (21)$$

In order to implement the algorithm, we approximate differentials by finite differences using the forward Euler integration scheme, so that (16) becomes

$$\mathbf{G}_k = -\text{tril}[\mathbf{W}_k\mathbf{R}\mathbf{W}_k^T]\mathbf{W}_k\mathbf{R} \quad (22)$$

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu\left[\mathbf{G}_k\mathbf{W}_k^T - \mathbf{W}_k\mathbf{G}_k^T\right]\mathbf{W}_k + \mu\mathbf{W}_k(\mathbf{W}_k^T\mathbf{W}_k\mathbf{R})[\mathbf{I} - \mathbf{W}_k^T\mathbf{W}_k] \quad (23)$$

Here, μ is a constant step size value.

Fig. 1 shows the convergence of the three errors

$$E_k^{(d)} = \frac{||\mathbf{W}_k\mathbf{R}\mathbf{W}_k^T - \mathbf{I}\odot\mathbf{W}_k\mathbf{R}\mathbf{W}_k^T||_F^2}{||\mathbf{I}\odot\mathbf{W}_k\mathbf{R}\mathbf{W}_k^T||_F^2} \quad (24)$$

$$E_k^{(p)} = \frac{||\mathbf{\Lambda}^{(p)} - \mathbf{I}\odot\mathbf{W}_k\mathbf{R}\mathbf{W}_k^T||_F^2}{||\mathbf{\Lambda}^{(p)}||_F^2} \quad (25)$$

$$E_k^{(o)} = ||\mathbf{I} - \mathbf{W}_k\mathbf{W}_k^T||_F^2, \quad (26)$$

where $\odot$ denotes element-by-element multiplication of matrix entries and $\mathbf{R}$ is given by

$$\mathbf{R} = \begin{bmatrix} 2.4 & 2.0 & 0.5 & 0.1 & 1.7 \\ 2.0 & 9.6 & 1.5 & 3.8 & 2.7 \\ 0.5 & 1.5 & 1.5 & 0.3 & 0.6 \\ 0.1 & 3.8 & 0.3 & 2.7 & 0.2 \\ 1.7 & 2.7 & 0.6 & 0.2 & 2.2 \end{bmatrix}, \quad (27)$$

$\mu = 0.005$, $m = 3$, $n = 5$, and $\mathbf{\Lambda}^{(p)}$ is a diagonal matrix containing the m largest eigenvalues in decreasing magnitude along its diagonal. The first error $E_k^{(d)}$ is small when $\mathbf{W}_k$ diagonalizes the matrix $\mathbf{R}$ in any m-dimensional subspace. The second error $E_k^{(p)}$ is small when the rows of $\mathbf{W}_k$ are aligned in order with the eigenvectors of the m largest eigenvalues of $\mathbf{R}$. The third error $E_k^{(o)}$ is small when $\mathbf{W}_k$ has orthonormal rows. All three errors must be small to solve the problem stated in the first paragraph of this section. As can be seen, all three errors converge to MATLAB's machine precision; the algorithm works as desired. Note that such an algorithm would be difficult and perhaps impossible to build with the Stiefel manifold update in (17) alone, as finding a single inhomogeneous cost whose maximization would yield both the m-dimensional principal subspace and the ordered eigenvectors within this subspace might be challenging.

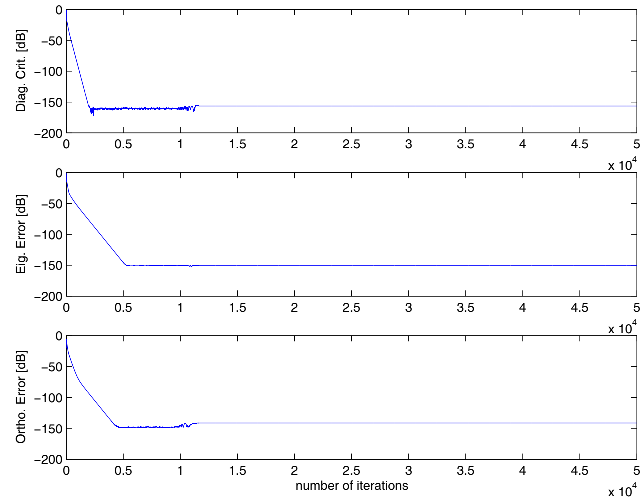

Fig. 1: Evolutions of the errors in the first simulation example.

A simple modification allows us to find the eigenvectors corresponding to the m smallest eigenvalues of $\mathbf{R}$ in the order of their increasing magnitudes. Consider the cost functions

$$\mathcal{J}_S(\mathbf{W}) = -\frac{1}{2}||\text{triu}[\mathbf{W}\mathbf{R}\mathbf{W}^T]||_F^2 \quad (28)$$

$$\mathcal{J}_G(\mathbf{W}) = -\frac{1}{2}\text{tr}[\mathbf{W}\mathbf{R}\mathbf{W}^T], \quad (29)$$

where triu$[\mathbf{M}]$ denotes the strictly upper triangular portion of the square matrix $\mathbf{M}$. The design of the discretized approximation to (16) must be done with care due to numerical issues associated with the Grassmann portion of the update, as is described in [3]. The corresponding algorithm is

$$\mathbf{G}_k = -\text{triu}[\mathbf{W}_k\mathbf{R}\mathbf{W}_k^T]\mathbf{W}_k\mathbf{R} \quad (30)$$

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu\left[\mathbf{G}_k\mathbf{W}_k^T - \mathbf{W}_k\mathbf{G}_k^T\right]\mathbf{W}_k - \mu\left[\mathbf{W}_k\mathbf{W}_k^T\mathbf{W}_k\mathbf{W}_k^T\mathbf{W}_k\mathbf{R} - \mathbf{W}_k\mathbf{R}\mathbf{W}_k^T\mathbf{W}_k\right] \quad (31)$$

Figure 2 shows the convergence of the three errors $E_k^{(d)}$, $E_k^{(m)}$, and $E_k^{(o)}$ for $\mu = 0.005$, $m = 3$, $n = 5$, and $\mathbf{R}$ as in

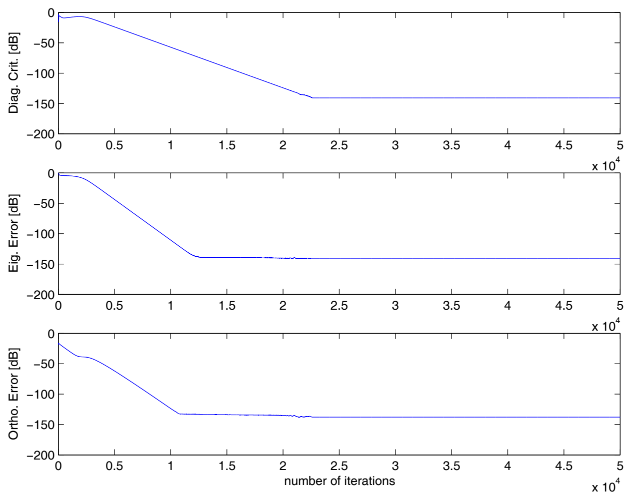

Fig. 2: Evolutions of the errors in the second simulation example.

(27), where

$$E_k^{(m)} = \frac{||\mathbf{\Lambda}^{(m)} - \mathbf{I} \odot \mathbf{W}_k \mathbf{R} \mathbf{W}_k^T||_F^2}{||\mathbf{\Lambda}^{(m)}||_F^2} \qquad (32)$$

and $\mathbf{\Lambda}^{(m)}$ is a diagonal matrix containing the m smallest eigenvalues in increasing magnitude along its diagonal. All three errors must be small to solve the problem stated in the first paragraph of this section. As can be seen, all three errors converge to MATLAB's machine precision; the algorithm works as desired.

A similar algorithm to that in (31) could be developed using the Stiefel manifold differential update in (17) due to the similarities of the cost functions in (28) and (29), respectively. Such an algorithm could not be easily redesigned to solve the principal eigenvector task or even to achieve smaller goals such as changing the order in which the eigenvectors appear within the rows of $\mathbf{W}_k$, however. The update in (16) has additional freedom in this regard.

5. MITIGATING NUMERICAL EFFECTS

As is well-known, discretization errors can cause the set of parameters within a tangent gradient algorithm to diverge away from its respective manifold. This fact has led numerous researchers to consider methods to approximate true geodesic motion in such algorithms using Taylor series expansions, correction terms, and the like in discretized coefficient updates. Interestingly, the algorithms developed in this way can look similar to our combined Grassmann-Stiefel update in (16), in which the second Grassmann-like term is introduced to reduce the error propagation of the parameter set away from the manifold. Adding such terms increases the complexity of an algorithm for the sake of numerical robustness without giving any additional estimation capability in return.

The creation of a combined Grassmann-Stiefel update partitions the problem of numerical error propagation in a nice way. Clearly, the error propagation of the algorithm away from the manifold is determined, and therefore controlled, by the Grassmann portion of the update. All of these numerical issues can and should be solved by modification of this portion of the algorithm, leaving the Stiefel part unmodified. The combined Grassmann-Stiefel update gives us an additional capability in return by allowing us to consider whatever goal we wish to achieve in the subspace estimation portion of the overall task. These ideas have been used in the two examples in the previous section, whereby numerically-stable principal and minor subspace tracking updates have been employed in these algorithm portions [1, 3]. Thus, the additional computations needed for numerical stability can be used to our advantage.

6. CONCLUSIONS

In this paper, we have considered a unified Grassmann-Stiefel update for adjusting an $(m \times n)$ orthonormal matrix within the Stiefel manifold to solve signal processing tasks. The new algorithmic approach allows us to partition the overall task into two parts: (1) m-dimensional subspace estimation within n-dimensional Euclidean space, and (2) the estimation of the coordinate system within the m-dimensional subspace. Different criteria can be used within each part, allowing for a rich set of algorithms to be developed. Numerical error propagation away from the Stiefel manifold is encapsulated within and can be controlled by the Grassmann portion of the update. Algorithms have been developed for principal and minor eigenvector estimation tasks. The insights gained from this study add to the ever-increasing body of literature devoted to geometric optimization methods.

7. REFERENCES

[1] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *J. Math Anal. Appl.,* vol. 106, no. 1, pp. 69-84, 1985.

[2] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing,* vol. 44, pp. 3017-3030, Dec. 1996.

[3] S.C. Douglas, S.-Y. Kung, and S. Amari, "A self-stabilized minor subspace rule," *IEEE Signal Processing Lett.,* vol. 5, pp. 328-330, Dec. 1998.

[4] A. Edelman, T.A. Arias, and S.T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal., Appl.* vol. 20, pp 303-353, Apr. 1999.

[5] S.C. Douglas, "Self-stabilized gradient algorithms for blind source separation with orthogonality constraints," *IEEE Trans. Neural Networks,* vol. 11, no. 6, pp. 1490-1497, Nov. 2000.

[6] J.H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Trans. Signal Processing,* vol. 50, pp. 635-650, Mar. 2002.

[7] M.D. Plumbley, "Lie group methods for optimization with orthogonality constraints," *Proc. 5th Int. Conf. Indep. Compon. Anal. Blind Signal Separation,* Granada, Spain, vol. 3195, pp. 1245-1252, Oct. 2004.

[8] J.H. Manton, U. Helmke, and I.M.Y. Mareels, "A dual purpose principal and minor component flow," *Systems and Control Lett.,* vol. 54, pp. 759-769, 2005.

[9] T. Abrudan, J. Eriksson, and V. Koivunen, "Optimization under unitary matrix constraint via multiplicative update," preprint.

[10] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis* (New York: Wiley, 2001).

[11] Y. Hua, "Asymptotical orthonormalization of subspace matrices without square root," *IEEE Signal Processing Mag.,* vol. 21, no. 4, pp. 56-61, July 2004.

[12] S.C. Douglas, "On the singular value manifold and numerical stabilization of algorithms with orthogonality constraints," *Proc. IEEE Workshop Sensor Array Multichannel Signal Processing,* Boston, MA, pp. 195-119, July 2006.